

FIG. 1

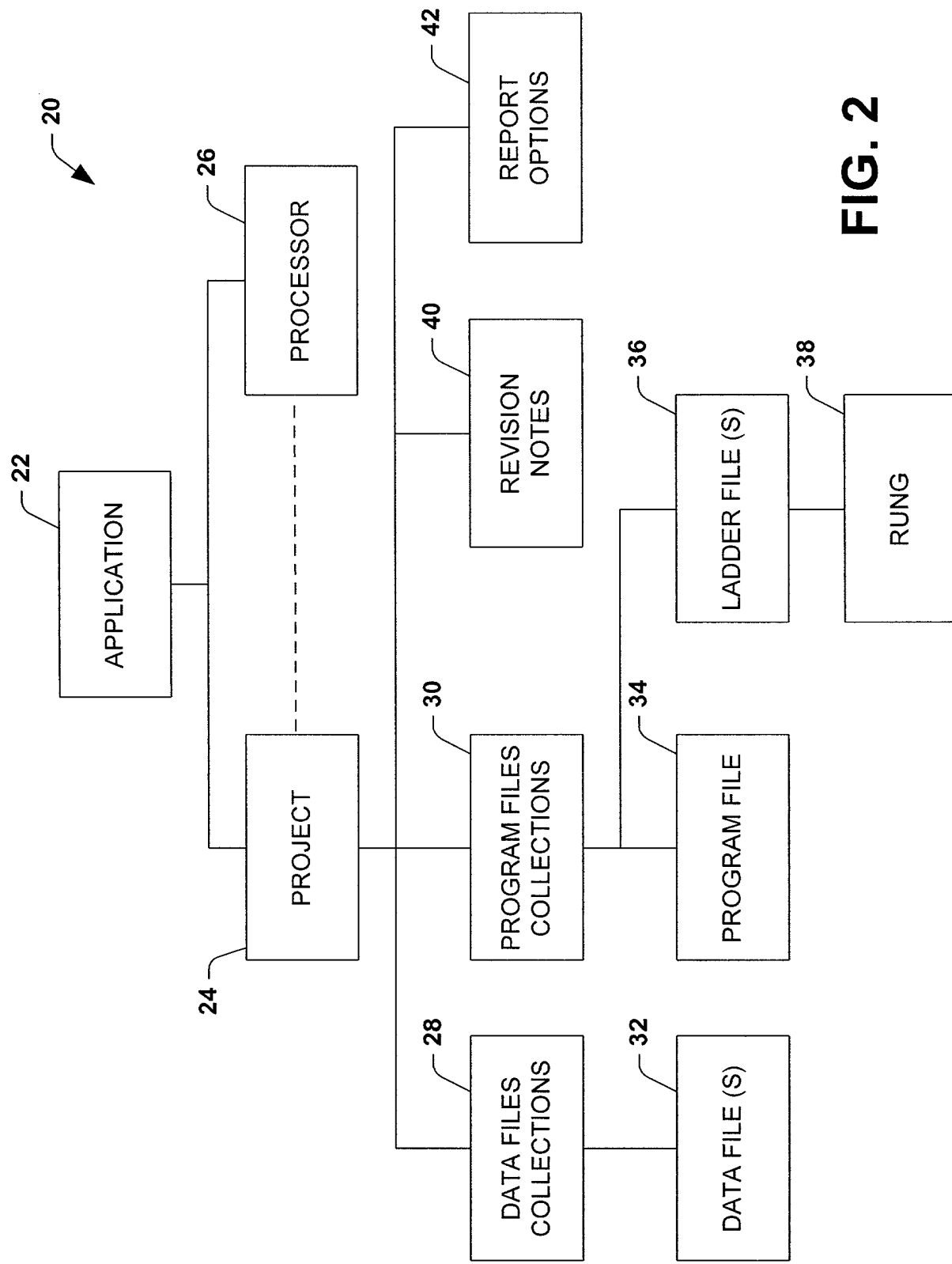


FIG. 2

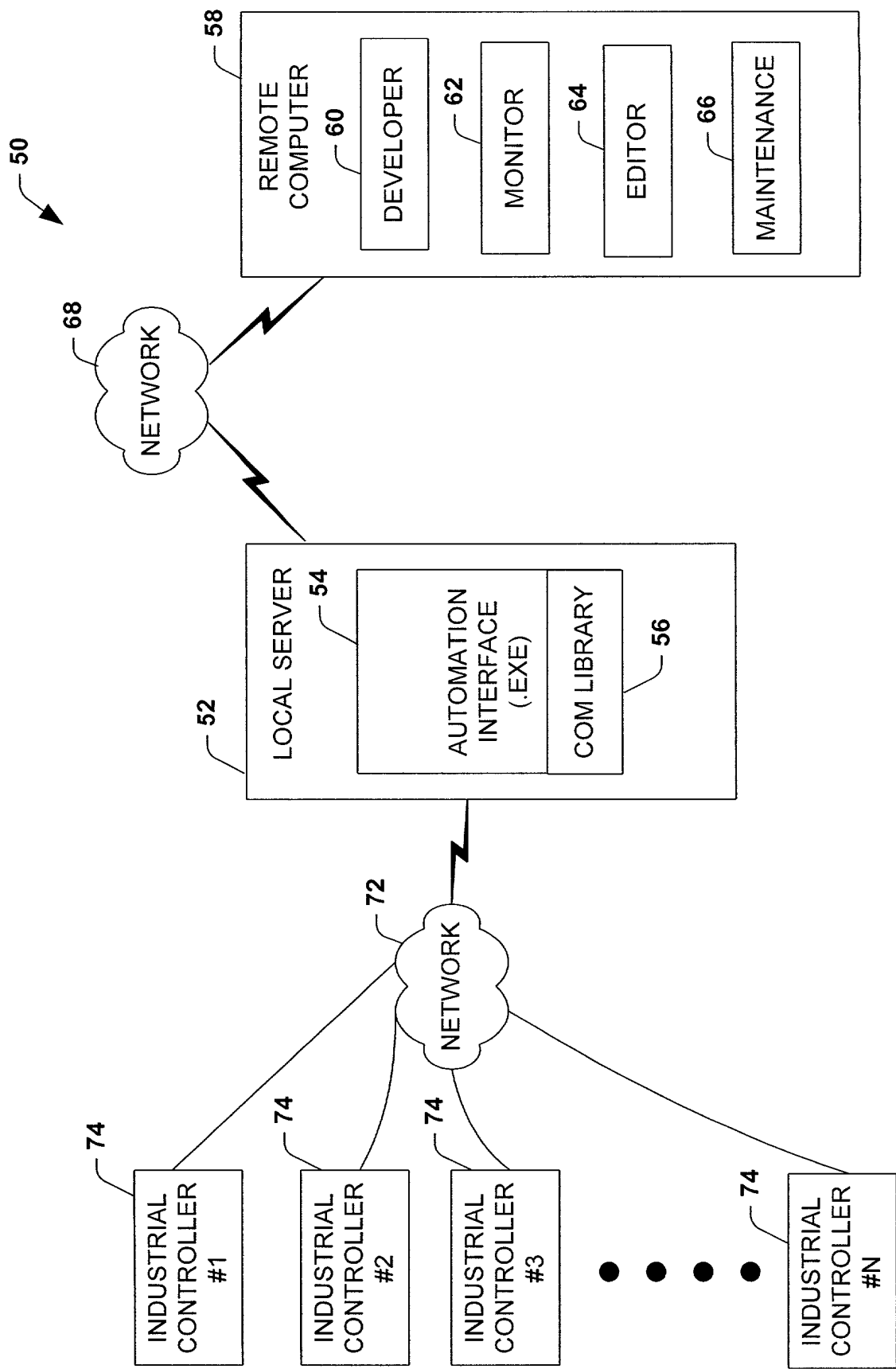


FIG. 3

FIG. 4 is a block diagram of a system 80 for providing a client application 84 with data from a data warehouse 90 via the Internet 88. The system 80 includes a client application 84, a web service/browser 82, a web service 92, an automation interface 94, an industrial controller(s) 96, and a data warehouse 90. The client application 84 is connected to the web service/browser 82. The web service/browser 82 is connected to the Internet 88. The Internet 88 is connected to the web service 92. The web service 92 is connected to the automation interface 94. The automation interface 94 is connected to the industrial controller(s) 96. The industrial controller(s) 96 is connected to the data warehouse 90. The data warehouse 90 is connected to the Internet 88.

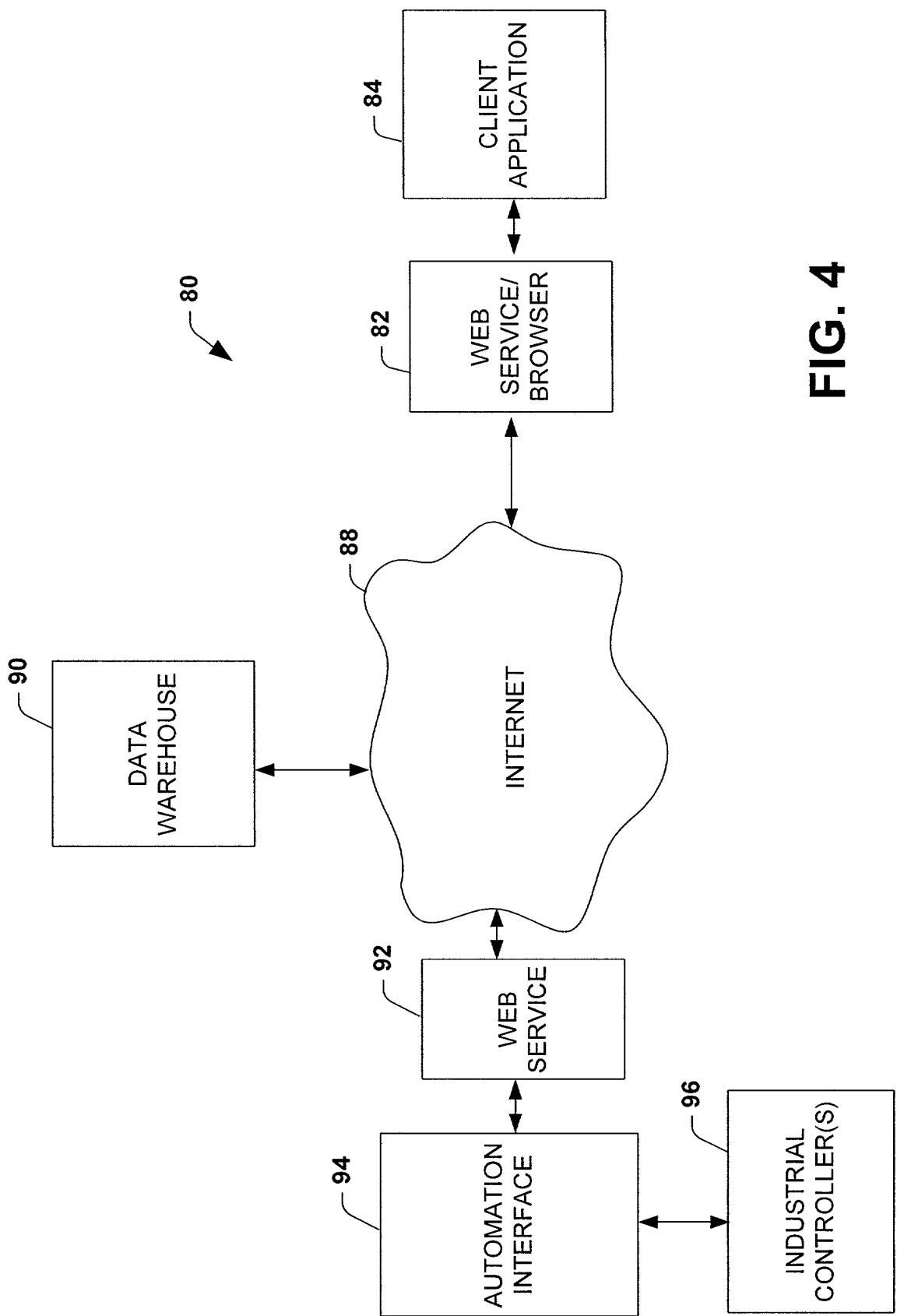


FIG. 4

```
01 Public Sub DownloadAProcessor()  
02     Dim g_Application As AutoInterface.Application  
03     Dim g_Project As AutoInterface.Project  
04     Dim EncodedRouteData As String  
05  
06     '** start Project and store it in the g_Application object variable  
07     Set g_Application = CreateObject("Project.Application")  
08  
09     '** Declare some variables for clarity.  
10     Dim ShowDialog As Boolean  
11     Dim UseAutoSave As Boolean  
12     Dim IgnorePrompts As Boolean  
13     Dim OnlineAction As IgxOnlineAction  
14  
15     '** Initialize these variables to suitable defaults  
16     ShowDialog = False  
17     UseAutoSave = False  
18     IgnorePrompts = True  
19     OnlineAction = IgxGoOffline  
20  
21     '** Open a project from disk that will be downloaded.  
22     Set g_Project = g_Application.FileOpen( "C:\Projects\Upload.rsp", ShowDialog, UseAutoSave)  
23  
24     '** The communication route to the actual processor that this project represents is set up in the  
25     '** application when the project is opened. Get this info out now and save it away for later calls  
26     '** to upload this processor.  
27     EncodedRouteData = g_Project.EncodedRouteString  
28     SaveEncodedRouteData EncodedRouteData  
29  
30     '** Now download the image to the processor.  
31     Dim NoError As Boolean  
32     NoError = g_Project.Download(IgnorePrompts, OnlineAction, IgxREMOTEPROG)  
33 End Sub
```

FIG. 5

```

01 Public Sub UploadAProcessor()
02     Dim g_Application As AutoInterface.Application
03     Dim g_Project As AutoInterface.Project
04     Dim EncodedRouteData As String
05
06
07
08     '** start Project and store it in the g_Application object variable
09     Set g_Application = CreateObject("Project.Application")
10
11     '** EncodedRouteData holds a string of data that points the project software to the processor on
12     '** the Data Highway+ network it will be performing uploads and downloads on. A function is
13     '** called here to initialize it to a previously obtained value
14
15     InitializeRouteData EncodedRouteData
16
17     '** Now tell project which processor it is to communicate with
18     g_Application.EncodedRouteString = EncodedRouteData
19
20     '** Declare some variables for clarity.
21     Dim IgnorePrompts As Boolean
22     Dim SaveChanges As Boolean
23     Dim AcceptDefaultAction As Boolean
24     Dim UploadAction As IgxUploadDownloadAction
25     Dim OnlineAction As IgxOnlineAction
26     Dim DataBaseAction As IgxSaveAction
27
28     '** Initialize these variables to suitable defaults
29     IgnorePrompts = True
30     SaveChanges = False
31     AcceptDefaultAction = True
32     UploadAction = IgxUploadCurrent
33     OnlineAction = IgxGoOffline
34     DataBaseAction = IgxNoAction
35
36     '** Now make the call that performs the upload. Store the uploaded project object
37     Set g_Project = g_Application.Upload(IgnorePrompts, SaveChanges, UploadAction,
38         _OnlineAction)
39
40     '** Now save the uploaded image to disk as a file called Upload.rsp
41     Dim NoError As Boolean
42     NoError = g_Project.SaveAs(IgnorePrompts, AcceptDefaultAction, _
43         DataBaseAction, "C:\Projects\Upload.rsp")
44 End Sub

```

FIG. 6

```

01 Public Sub InsertLadderRung()
02 Dim g_Application As AutoInterface.Application
03 Dim g_Project As AutoInterface.LogixProject
04 Dim g_ProgFile As AutoInterface.ProgramFile
05 Dim g_LadderFile As AutoInterface.LadderFile
06
07 *** start AutoInterface and store it in the g_Application object variable
08 Set g_Application = CreateObject("AutoInterface.Application")
09
10 *** Declare some variables for clarity.
11 Dim ShowDialog As Boolean
12 Dim UseAutoSave As Boolean
13 Dim IgnorePrompts As Boolean
14 Dim AcceptDefaultAction As Boolean
15 Dim OnlineAction As IgxOnlineAction
16
17 *** Initialize these variables to suitable defaults
18 ShowDialog = False
19 UseAutoSave = False
20 IgnorePrompts = True
21 AcceptDefaultAction = True
22 OnlineAction = IgxGoOffline
23
24 *** Open a project from disk that will be modified.
25 Set g_Project = g_Application.FileOpen( "C:\Projects\Upload.rsp", ShowDialog, UseAutoSave)
26
27 *** Obtain the program file object for program file 2. Cast this object to a LadderFile object since
28 *** in this case, program file 2 is also a ladder file
29 Set g_ProgFile = g_Project.ProgramFiles(2)
30 Set g_LadderFile = g_ProgFile
31
32 *** Build up a sample rung manually
33 Dim RungString As String
34 RungString = "SOR XIC B3:0/0 OTE B3:0/1 EOR"
35
36 *** Insert this rung into the ladder at position 0
37 g_LadderFile.InsertRungAsAscii 0, RungString
38
39 *** Save the modified project to disk first, then
40 *** download the image to the processor and set the processor to RUN mode.
41 *** Once this is completed, the new rung is executing in the processor.
42 Dim NoError As Boolean
43 NoError = g_Project.Save(IgnorePrompts, AcceptDefaultAction)
44 NoError = g_Project.Download(IgnorePrompts, OnlineAction, IgxREMOTERUN)
45 End Sub

```

FIG. 7

130

Application Object

Start AUTOINF

Upload

GetActiveProject

Stop AUTOINF

Project Object

Save

Download

Project FullName Property

Project name Property

Processor Object

Processor Node

Enable Forces

Program Files Collection

Program File

Add

Remove

Get

For ProgramFile

Program File

Type

Name

Convert to LadderFile

LadderFile

Rung Number

Rung Ascii

Add Rung

Rung

Rung Number

Get

Rung Title

Rung Component

FIG. 8

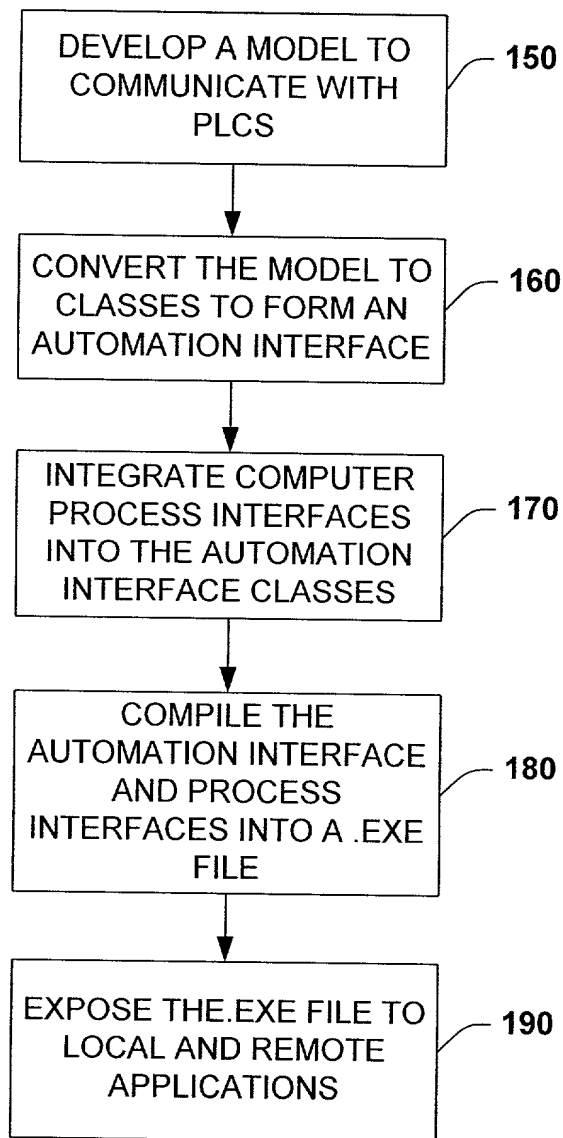


FIG. 9

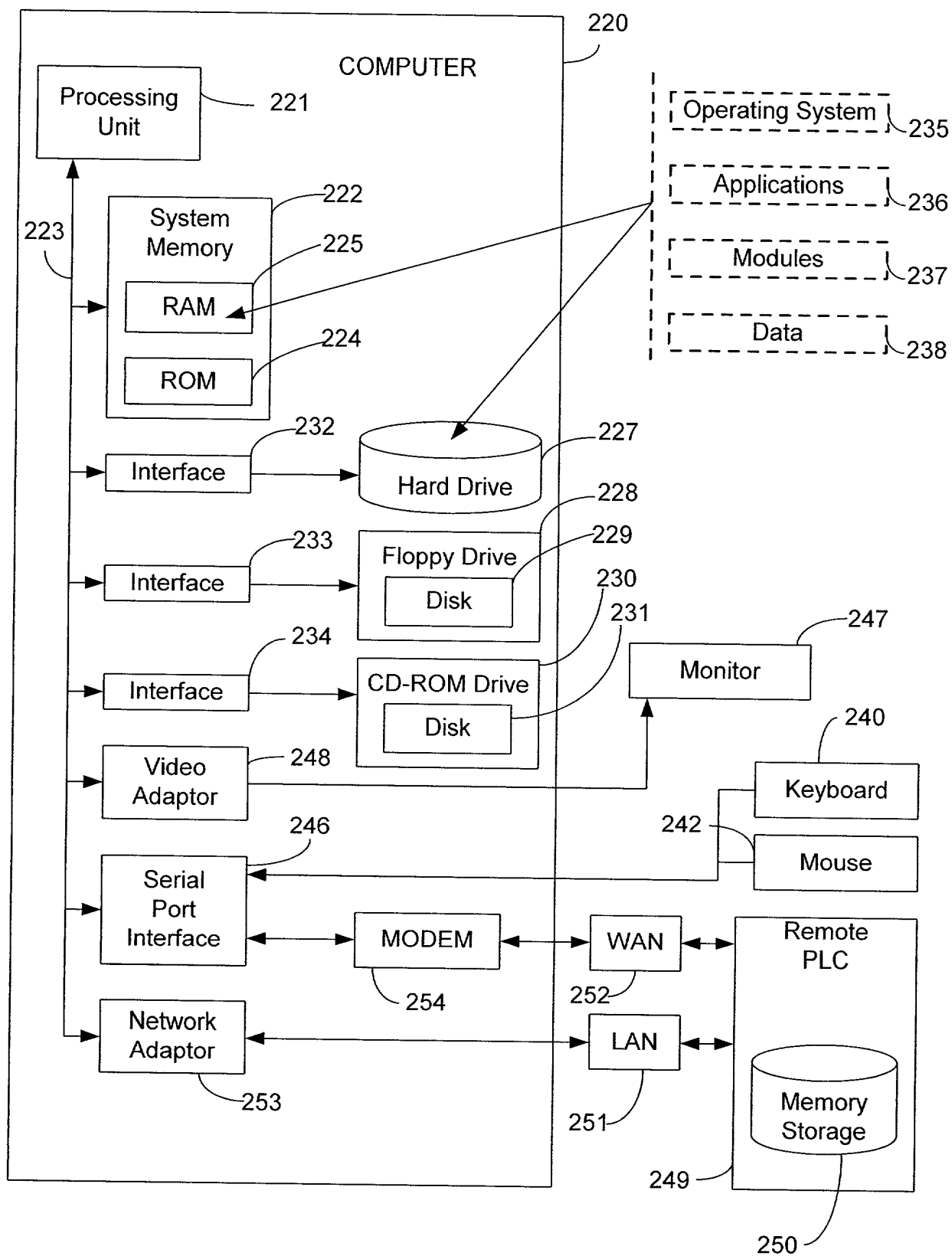


FIG. 10